Crisis detection from Arabic tweets

Alharbi, Alaa Ali H; Lee, Mark

Crisis Detection from Arabic Tweets

Alaa Alharbi[1,2] and Mark Lee[1]

[1]School of Computer Science, University of Birmingham, Birmingham, UK
[2]College of Computer Science and Engineering, Taibah University, Medina, KSA
alaharbi@taibahu.edu.sa & m.g.lee@cs.bham.ac.uk

Abstract

Social media (SM) platforms such as Twitter offer a rich source of real-time information about crises from which useful information can be extracted to support situational awareness. The task of automatically identifying SM messages related to a specific event poses many challenges, including processing large volumes of short, noisy data in real time. This paper explored the problem of extracting crisis-related messages from Arabic Twitter data. We focused on high-risk floods as they are one of the main hazards in the Middle East. In this work, we presented a gold-standard Arabic Twitter corpus for four high-risk floods that occurred in 2018. Using the annotated dataset, we investigated the performance of different classical machine learning (ML) and deep neural network (DNN) classifiers. The results showed that deep learning is promising in identifying flood-related posts.

1 Introduction

Social media (SM) platforms provide a valuable source of real-time information about emergency events. During mass emergencies, microblogging sites such as Twitter are used as communication channels by people and organisations to post situational updates, provide aid, request help and search for actionable information. Examples of Twitter's effectiveness during crises include the Manila floods in 2013 (Olteanu et al., 2015), the Louisiana flood in 2016 (Kim and Hastak, 2018) and tropical storm Cindy in 2017 (Kim et al., 2018). Twitter was used to report the protests that followed the Iranian presidential elections of 2009 (Khondker, 2011). It also played an important role in the Arab Spring (Arafa and Armstrong, 2016). For instance, Twitter was used as a means of communication by protesters during the Egyptian revolution in February 2011 (Tufekci and Wilson, 2012). Petrovic et al. (2013) found that Twitter often breaks incoming news about disaster-related events faster than traditional news channels. The early identification of disaster-related messages enables decision-makers to respond quickly and effectively during emergencies.

The huge volume of user-generated Twitter data related to numerous daily events has given rise to the need for automatic event extraction and summarising tools. Event extraction from Twitter streams poses challenges that differ from traditional media. In particular, traditional text extraction techniques are challenged by the noisy language used in social media, including colloquialisms, misspelled words and non-standard acronyms. Because of the imposed character limit (280 characters), Twitter users tend to use more abbreviations and may also post non-informative messages that require some knowledge of the situational context for interpretation. In addition, Twitter's popularity makes it appealing for spammers who spread propaganda, pornography and viruses (Benevenuto et al., 2010; Kabakus and Kara, 2017). Another challenge posed by Twitter is that the increasing volume and high-rate data stream of user-generated messages create significant computational demand.

Previous studies that have explored the problem of extracting crisis-related messages from SM have proposed various matching-based and learning-based approaches. Supervised machine learning (ML) and deep learning models have been used to identify event-relevant messages and classify them into several categories. A significant percentage of such studies have been conducted on English SM text. Very little work has focused on Arabic text. The Arabic language has its own peculiarities that make classifying Arabic SM text more challenging. For example, SM users sometimes write in their own dialects. There exist many spoken Arabic dialects that differ in their phonology, morphology and syntax (Chiang et al., 2006). People tend to write the dialectical words

according to their own pronunciations. There is no spelling standard for written dialectical words. Dialects are region-based. Hence, a classifier trained on data collected from one region may not perform well when tested on data collected from another region.

Unlike English, Arabic has poor available resources. To the best of our knowledge, there is no publicly available Arabic crisis-related dataset. We therefore built our own. In this work, we focused on flooding crises as they are a major hazard in the Middle East. A crisis usually occurs after heavy rain and subsequent flash flooding. In October and November 2018, heavy rainfall caused severe flooding in various Middle Eastern countries including Saudi Arabia, Kuwait, Jordan, Qatar, Iraq and Iran. According to civil defence authorities in Saudi Arabia, 1,480 individuals were rescued, 30 died and 3,865 were evacuated during floods that occurred in the period between 19 October and 14 November.[1] In Jordan, the flash flood on 9 November left at least 12 people dead and 29 injured.[2] On the same day, Kuwait had heavy rain that resulted in infrastructure and property damage and left at least one person dead.[2]

This research used different supervised learning approaches to extract flood-related tweets for the purpose of enhancing crisis management systems. We investigated the ways in which deep neural networks (DNN) compare to ML models in identifying crisis-related SM messages. Inspired by Nguyen et al. (2017), we also explored how different models perform when they are trained on historical event data, as labelling data from current events is expensive. Furthermore, continually retraining a model from scratch using data from current events is undesirable as it delays the timely processing of messages. The contributions of this paper as follows:

- We provide an annotated Arabic Twitter dataset of flood events.
- We benchmark the dataset using different supervised learning approaches.
- We evaluate the performance of two classical ML models and four DNNs on extracting flood-related messages, under two training settings: (1) train and test on the same event data; and (2) train on previous in-domain events and test on the current event.

The rest of this paper is organised as follows: section 2 surveys related work. Section 3 describes the process of building the Arabic flood Twitter dataset. Section 4 presents the used ML and DNNs models. The experimental settings and the results are detailed in sections 5 and 6, respectively. Finally, section 7 concludes the paper and discusses future work.

2 Related Work

A review of the recent literature confirms widespread interest in detecting and extracting information from Twitter posts that describe current events. Recently, extracting crisis-related events from social media has received considerable attention.

Kireyev et al. (2009) experimented with latent Dirichlet allocation (LDA) topic models to detect disasters from Twitter posts. Sakaki et al. (2010) developed an earthquake reporting system by processing Twitter data. They used a support vector machine (SVM) to classify Twitter messages into two groups: event and non-event. They also proposed temporal and spatial models to estimate the earthquake's location. Cameron et al. (2012) presented a model to detect crises from Twitter using burst detection and incremental clustering. Abel et al. (2012) described a framework called Twitcident for searching, filtering and analysing Twitter streams during incidents. Twitcident monitors broadcasting services and translates incident-related messages into profiles for use as Twitter search queries to extract relevant tweets.

Using a supervised ML approach, Imran et al. (2013a) classified Twitter posts into fine-grained classes and extracted the relevant information from the messages. In a subsequent work, they described a method for extracting disaster information using conditional random fields (CRF) (Imran et al., 2013b). Ashktorab et al. (2014) described a supervised learning-based approach to identifying disaster-related tweets and extracting actionable information.

Singh et al. (2017) developed a classification-based system to extract flood-related posts and classify them as high or low priority to identify victims who need urgent assistance. Nguyen et al. (2017) and Caragea et al. (2016) used convolutional neural network (CNN) to identify in-

[1] https://sabq.org/jGVvgZ

[2] http://floodlist.com/asia/jordan-flash-floods-november-2018

formative (useful) messages from crisis events. Nguyen et al. (2017) highlighted that CNN performed better than many classical ML approaches. Going further, Neppalli et al. (2018) compared the performance of a naïve Bayes (NB) classifier to two deep neural models in identifying informative crisis-related posts. Their results demonstrated that CNN outperformed both the recurrent neural network (RNN) with gated recurrent unit (GRU) model and the NB with handcrafted features. Unlike the described work, which focused on the classification of English tweets to extract the relevant event messages, Alabbas et al. (2017) used supervised ML classifiers to identify high-risk flood-related tweets that were written in Arabic.

Using different classical ML and deep learning approaches, we also classified the Arabic tweets as flood-related or irrelevant. Our work differs from that of Alabbas et al. (2017) in the classification techniques and data collection. Instead of tracking the Arabic words سيول، فيضانات (floods), we based our collection on event-related keywords as described in the following section.

3 Corpus Collection and Annotation

Using the Twitter API,[3] Arabic tweets were collected by tracking certain keywords and hashtags related to 10 flood events. The tracked floods occurred in the Middle East in October and November, 2018. The initial set of tweets for each event were crawled based on the event-related trendy hashtags or by searching for tweets containing the terms سيول (floods) and the flood location name. Then, the dataset were expanded by tracking all the relevant hashtags found in the collected set. This step was repeated until no new event-related hashtag could be found. Different numbers of messages were obtained per event. While we managed to crawl thousands of tweets for some events, we ended up with just a few hundred for others. The size of candidate flood-related data might depend on the popularity and severity of the event.

In this research, only four events were considered for annotation. The events were: Jordan floods, Kuwait floods, Qurayyat floods and Al-Lith floods. The selected events took place in different areas of the Arab world: Jordan, Kuwait, northern Saudi Arabia and western Saudi Arabia, respectively. Hence, we believed that the dataset should include tweets written in different Arabic dialects. In addition, each of these events trended on Twitter. We collected plenty of candidate flood-related tweets, at least 5,000 for each of the four crisis under consideration. The four floods led to property and infrastructure damage. Three of them left several people displaced or dead. Therefore, we assumed that the collected messages would convey different types of disaster-related actionable information.

To construct the dataset, we first extracted the tweet IDs and texts from the event-related JSON files obtained by the Twitter streaming API. Each retweet was replaced with the original text of the retweeted message. We removed duplicates (i.e., tweets that had exactly the same text). After that, a random sample of around 1,050 distinct tweets was selected from each event to be annotated by a human. As the Qurayyat flood had only 954 distinct messages, we labelled them all. The corpus was annotated by four native Arabic speakers. They were provided with the annotation instructions, examples of ten labelled tweets and a brief description of each event. Annotators were asked to provide the appropriate label based on the tweet's text; they were not required to open any included hyperlinks. Each tweet was judged by two annotators who selected the most suitable label for the two tasks described below.

1. **Relevance:** The first task was to decide whether a message was on-topic/event-related or off-topic/not related. Very short and understandable messages that did not convey any meaning, such as those that only included hashtags, were ignored.

2. **Information type:** In order to build classifiers that could identify informative crisis-related messages, tweets that communicated useful information were labelled based on the information category they provided. This task followed the annotation scheme described by Olteanu et al. (2015), which labelled each message as one of the following broad categories:

    - Affected individuals: included reports on affected, dead, missing, trapped, found or displaced people
    - Infrastructure and utilities damage

[3] https://help.twitter.com/en/rules-and-policies/twitter-api

- Donations, assistance and volunteering services
- Caution and advice
- Sympathy, prayers and emotional support
- Other useful information: messages that did not belong to the previous categories but helped in understanding the emergency situation
- Not applicable: the message was either irrelevant or did not communicate any useful information, e.g., personal opinions.

We measure inter-rater agreement with Cohen's Kappa, resulting in $k \approx 0.82$ for relevance and $k \approx 0.9$ for information type. In cases when the two annotators disagreed, the tweet was judged by a third person. The final dataset[4] included 4,037 labelled Twitter messages for four flood events. Table 1 presents a general description of the dataset along with the number of relevant and irrelevant messages per event. In our corpus, 24.69% of tweets were irrelevant. Table 2 shows the distribution of information categories per event.

4 Models

4.1 Classical ML Models

The performance of classic ML models depends mainly on how the features are extracted and selected. To benchmark the dataset, we experimented with SVM and NB for flood-related message identification.

4.2 Deep Learning Models

Deep learning has profound generalisation ability and has proven to perform well in text classification, achieving state-of-the-art results on standard natural language processing (NLP) benchmark problems. In this research, we experiment with the following deep learning models:

- Convolutional Neural Network (CNN): The network architecture was similar to that proposed by Kim (2014). We used two 1D convolutions that were applied in parallel to the input layer vectors, extracting local patches from sequences using convolution windows of sizes 3 and 5 with 100 feature maps each. A sliding max-pooling operation of size 2 was applied over each feature map to obtain the maximum value, representing the most important feature. The output vectors of the two convolutions were concatenated and a 0.5 dropout rate was applied for regularisation. The output was fed into a 100-dimension fully connected layer with rectified linear unit (ReLU) activation.

- Long Short-Term Memory (LSTM): LSTM (Hochreiter and Schmidhuber, 1997) is a type of recurrent neural network (RNN) that can learn over long input sequences. In our experiments, this model involved one LSTM layer with 196 hidden output dimensionalities. As proposed by Gal (2016), we applied a dropout rate for input units of the LSTM layer and a dropout rate of the recurrent units for regularisation. In the experiments, both were set to 0.2.

- Convolution LSTM (CLSTM): This model is similar to the CNN described above except that the fully connected dense layer is replaced by an LSTM layer similar to the one presented above. In this architecture, the CNN was used to extract features that were fed into an LSTM layer, which processed down-sampled high-level input sequences.

- Bidirectional LSTM (BiLSTM): BiLSTM is a another type of RNN. In processing input sequences in both forward and backward directions, BiLSTM merges their representations to capture patterns that might be missed by order-dependent RNNs such as LSTM. The bidirectional model in our experiments had 196 hidden dimensions and dropout rates equal to the ones used in the LSTM model.

The input sequences, the embedding and output layers were similar for all previously described DNN models. The embedding layer was used as the first hidden layer to map words (input sequences) to dense vectors. In our experiments, vectors were initialised from an external embedding model and fine-tuned during training. The output layer mapped its input vectors – which were obtained from the last hidden layer in each model – to a probability between 0 and 1 using the sigmoid activation function.

[4] It is available for research purposes at https://www.cs.bham.ac.uk/~axa1314/

Crisis	Country	# of Labelled Posts	# of On-topic Posts	# of Off-topic Posts
Jordan floods	Jordan	1009	761	248
Kuwait floods	Kuwait	1056	822	234
Qurayyat floods	Saudi	954	705	249
Al-Lith floods	Saudi	1018	752	266

Table 1: Dataset Description.

Flood Name	Affected individuals	Infrastructure & utilities	Donations & volunteering	Caution & advice	Sympathy & emotional support	Other useful information	Not applicable
Jordan	200	60	15	79	269	76	310
Kuwait	96	106	30	62	191	53	518
Qurayyat	184	58	8	244	131	37	292
Al-Lith	70	196	81	212	72	58	329

Table 2: Distribution of tweets by information types.

5 Experiments

In this study, we performed a binary classification task to identify flood-related messages. Identifying the information category of the relevant tweets is left for future work. As the dataset had imbalanced classes, we first up-sampled the minority class to have a relatively equal class distribution. Then we preprocessed the tweets as described below.

5.1 Text Preprocessing:

To improve model generalisation, we replaced each URL with the Arabic word رابط (hyperlink). In the same way, each user handle was substituted with the word مستخدم (username), while numbers were replaced with the word رقم (number). We also normalised repeated letters and elongation (Tatweel). Diacritics or short vowels, non-Arabic characters, punctuation and special characters were removed. We performed three types of letter normalisations: the variant forms of *alef* (أ، إ، آ) were normalised to (ا) , *alef maqsora* (ى) to *ya* (ي) and *ta marbouta* (ة) to *ha* (ه) . This was done because people often misspell various forms of *alef* and do not distinguish between *ta marbouta* and *ha* when these letters occur at the ends of words. In addition, stop words were removed. While stop word removal is not useful for some NLP tasks such as sentiment analysis, it can enhance the performance of some classification tasks as they do not affect the overall topic/meaning of a document. Finally, tweets were tokenised using the CMU Tweet NLP tool (Gimpel et al., 2010). We did not apply stemming to the tokens, as the previous work confirmed that stemming does not improve classification accuracy (Alabbas et al., 2017).

With respect to classic ML models, unigrams, bigrams and trigrams of words were extracted. In case of NB, text was represented as bags of words as we experimented with multinomial NB classifier which is suitable for classification with discrete features. For SVM model, the features were transformed into term-frequency inverse-document-frequency (TF-IDF) vectors, in which each tweet represented a document. For DNN models, texts were segmented into words. The maximum length of input sequences per tweet was set to 60 words. Messages comprising fewer than 60 words were zero-padded. Each word was transformed into a vector. Word vectors were initialised from Ara Vec (Soliman et al., 2017). Ara Vec was trained on Arabic Twitter text of 1,090 million tokens using a continuous bag of words (CBOW) technique with a window size of three words. In both types of models, we limited the vocabulary to the most common 5,000 words in the training corpus.

5.2 Training Settings:

We first examined the performance of the learning models in identifying the relevant messages when they were trained using data from the same event. In this case, the data was split into subsets of 80% for training and 20% for testing using 5-fold cross validation. Assuming that labelled data were not available for the current event, the second experiment evaluated the models' performance when they were trained using the historical events. Here, the entirety of the data pertaining to the event under consideration was used for training and testing.

5.3 Models' Settings:

Classic ML classifiers were implemented using the scikit-learn library (Pedregosa et al., 2011). We experimented with linear kernel SVM and multinomial NB classifiers. Deep learning models were built using the Keras library.[5] The DNN models were trained for 10 epochs in mini-batches of 10 samples. The optimiser and loss function arguments were set to adam and binary crossentropy, respectively.

6 Results

Table 3 shows the average accuracy scores for the first experiment, in which classifiers were trained on the event data using 5-fold cross-validation. The table indicates that DNNs performed very well despite the relatively small training dataset. The deep learning models yielded comparable performance. RNNs outperformed the ML models in all cases. LSTM and BiLSTM achieved the best accuracy scores. SVM returned results that were competitive with DNN models. Looking at the classification errors of the LSTM and BiLSTM models, we found that the most common error is the incorrect classification of minority class (off-topic tweets). This is due to the imbalanced dataset. The random over-sampling can increase the likelihood of overfitting the data as it creates exact copies of existing instances. We also found that some of the uninformative flood-related messages were mistakenly classified as off-topic. For instance, 14% of such messages in Kuwait data were incorrectly classified by the LSTM model.

As the identification of crisis-related messages is a time-critical task, it is unlikely to obtain sufficient labelled data from the current event. Hence, we explored how the classifiers perform in detecting relevant posts from different events within the same domain. The accuracy scores are displayed in Table 4. BiLSTM achieved the best accuracy in most cases. CLSTM and LSTM showed competitive results in certain instances. LSTM outperformed the CNN in 9 out of 11 experiments. This showed that RNNs could be more suitable to address such problems as they represent the whole input sequence instead of relying on some key local features. Feeding the extracted CNN features into an LSTM layer instead of a fully connected dense layer resulted in improved accuracy when training on one event. The structure of RNNs allows such models to learn problem-specific information about the mapping they approximate, which could reduce the training data requirement. As the number of training examples increased, CNN achieved performance comparable with CLSTM. Table 4 shows that SVM generalised better than NB model. Generally, it can be seen that DNNs outperformed the traditional ML models. DNNs use distributed representation of words and learn high-level abstract features (Imran et al., 2018). On the other hand, ML models' performance depends on the training data and manually engineered features and therefore perform poorly when tested in different crises due to the great variation of data.

In the first six cases, we trained the models using data from a single event. Taking chronological order into account, we then increased the number of events in the training set to see whether this could enhance performance. It could be noticed that all models showed the best accuracy in classifying Al-Lith messages when three events were used for training. However, increasing the number of training events did not always result in improved accuracy. For example, training DNNs using data from Kuwait and Qurayyat resulted in lower performance compared to the case when only Qurayyat data was used to classify Al-Lith messages. Similarly, the results acheived by using Jordanian data to train ML models were higher than those obtained by using the joint dataset of Jordan and Kuwait.

7 Conclusion

This paper investigated the problem of extracting flood-related data from Arabic tweets using a supervised learning approach. To the best of our knowledge, it is the first work that uses deep learning to identify crisis-related data from Arabic tweets. Our results show that RNNs are promising in identifying crisis messages using training data from the event or from other in-domain events. We also provided a gold-standard Arabic Twitter dataset for high-risk floods. For future work, we aim to evaluate the same models in multiclass identification to extract information types from flood-related messages. We also plan to utilise domain adaptation approaches to enhance the results of classifiers trained using data within the crisis domain.

[5] https://keras.io/

Event	SVM	NB	CNN	LSTM	CLSTM	BiLSTM
Jordan floods	91.03	79.72	91.77	91.26	91.69	**92.14**
Kuwait floods	89.45	83.76	90.58	**91.91**	89.87	91.21
Qurayyat floods	94.18	90.19	92.87	95.17	94.64	**95.48**
Al-Lith floods	90.83	81.59	93.86	**94.08**	91.64	93.56

Table 3: The accuracy scores of classical ML and DNN models when they are trained on event data.

Train Set	Test Set	SVM	NB	CNN	LSTM	CLSTM	BiLSTM
Jordan floods	Kuwait floods	61.60	63.15	67.51	70.32	67.01	**70.46**
Jordan floods	Qurayyat floods	68.42	56.47	70.95	72.10	71.03	**72.33**
Jordan floods	Al-Lith floods	**71.22**	69.23	64.49	65.82	67.75	71.07
Kuwait floods	Qurayyat floods	**69.73**	59.15	62.22	64.13	67.66	**69.73**
Kuwait floods	Al-Lith floods	63.90	61.98	67.15	67.30	**71.81**	71.59
Qurayyat floods	Al-Lith floods	68.19	68.04	75.22	**76.40**	75.66	76.03
Jordan + Kuwait floods	Qurayyat floods	69.80	60.30	73.10	75.24	73.02	**76.55**
Jordan + Kuwait floods	Al-Lith floods	69.89	68.04	71.30	68.93	71.59	**74.40**
Jordan + Qurayyat floods	Al-Lith floods	73.89	72.04	75.88	75.36	76.62	**77.73**
Kuwait + Qurayyat floods	Al-Lith floods	70.85	70.71	72.63	**75.88**	74.92	74.48
Jordan + Kuwait + Qurayyat floods	Al-Lith floods	75.51	72.11	76.84	**77.95**	77.81	77.66

Table 4: The accuracy scores of classical ML and DNN models when they are trained on out-of-event data.

References

Fabian Abel, Claudia Hauff, Geert-Jan Houben, Richard Stronkman, and Ke Tao. 2012. Twitcident: fighting fire with information from social web streams. In *Proceedings of the 21st International Conference on World Wide Web*, pages 305–308. ACM.

Waleed Alabbas, Haider M al Khateeb, Ali Mansour, Gregory Epiphaniou, and Ingo Frommholz. 2017. Classification of colloquial arabic tweets in real-time to detect high-risk floods. In *2017 International Conference On Social Media, Wearable And Web Analytics (Social Media)*, pages 1–8. IEEE.

Mohamed Arafa and Crystal Armstrong. 2016. " facebook to mobilize, twitter to coordinate protests, and youtube to tell the world": New media, cyberactivism, and the arab spring. *Journal of Global Initiatives: Policy, Pedagogy, Perspective*, 10(1):6.

Zahra Ashktorab, Christopher Brown, Manojit Nandi, and Aron Culotta. 2014. Tweedr: Mining twitter to inform disaster response. In *ISCRAM*.

Fabricio Benevenuto, Gabriel Magno, Tiago Rodrigues, and Virgilio Almeida. 2010. Detecting spammers on twitter. In *Collaboration, electronic messaging, anti-abuse and spam conference (CEAS)*, volume 6, page 12.

Mark A Cameron, Robert Power, Bella Robinson, and Jie Yin. 2012. Emergency situation awareness from twitter for crisis management. In *Proceedings of the 21st International Conference on World Wide Web*, pages 695–698. ACM.

Cornelia Caragea, Adrian Silvescu, and Andrea H Tapia. 2016. Identifying informative messages in disaster events using convolutional neural networks. In *International Conference on Information Systems for Crisis Response and Management*, pages 137–147.

David Chiang, Mona Diab, Nizar Habash, Owen Rambow, and Safiullah Shareef. 2006. Parsing arabic dialects. In *11th Conference of the European Chapter of the Association for Computational Linguistics*.

Yarin Gal. 2016. *Uncertainty in deep learning*. Ph.D. thesis, PhD thesis, University of Cambridge.

Kevin Gimpel, Nathan Schneider, Brendan O'Connor, Dipanjan Das, Daniel Mills, Jacob Eisenstein, Michael Heilman, Dani Yogatama, Jeffrey Flanigan, and Noah A Smith. 2010. Part-of-speech tagging for twitter: Annotation, features, and experiments. Technical report, Carnegie-Mellon Univ Pittsburgh Pa School of Computer Science.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Muhammad Imran, Carlos Castillo, Fernando Diaz, and Sarah Vieweg. 2018. Processing social media messages in mass emergency: Survey summary. In *Companion of the The Web Conference 2018 on The Web Conference 2018*, pages 507–511. International World Wide Web Conferences Steering Committee.

Muhammad Imran, Shady Elbassuoni, Carlos Castillo, Fernando Diaz, and Patrick Meier. 2013a. Extracting information nuggets from disaster-related messages in social media. In *Iscram*.

Muhammad Imran, Shady Elbassuoni, Carlos Castillo, Fernando Diaz, and Patrick Meier. 2013b. Practical extraction of disaster-relevant information from social media. In *Proceedings of the 22nd International Conference on World Wide Web*, pages 1021–1024. ACM.

Abdullah Talha Kabakus and Resul Kara. 2017. A survey of spam detection methods on twitter. *International Journal of Advanced Computer Science and Applications*, 8(3).

Habibul Haque Khondker. 2011. Role of the new media in the arab spring. *Globalizations*, 8(5):675–679.

Jooho Kim, Juhee Bae, and Makarand Hastak. 2018. Emergency information diffusion on online social media during storm cindy in us. *International Journal of Information Management*, 40:153–165.

Jooho Kim and Makarand Hastak. 2018. Social network analysis: Characteristics of online social networks after a disaster. *International Journal of Information Management*, 38(1):86–96.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*.

Kirill Kireyev, Leysia Palen, and Kenneth Anderson. 2009. Applications of topics models to analysis of disaster-related twitter data. In *NIPS Workshop on applications for topic models: text and beyond*, volume 1. Canada: Whistler.

Venkata Kishore Neppalli, Cornelia Caragea, and Doina Caragea. 2018. Deep neural networks versus naïve bayes classifiers for identifying informative tweets during disasters.

Dat Tien Nguyen, Kamela Ali Al Mannai, Shafiq Joty, Hassan Sajjad, Muhammad Imran, and Prasenjit Mitra. 2017. Robust classification of crisis-related data on social networks using convolutional neural networks. In *Eleventh International AAAI Conference on Web and Social Media*.

Alexandra Olteanu, Sarah Vieweg, and Carlos Castillo. 2015. What to expect when the unexpected happens: Social media communications across crises. In *Proceedings of the 18th ACM conference on computer supported cooperative work & social computing*, pages 994–1009. ACM.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. 2011. Scikit-learn: Machine learning in python. *Journal of machine learning research*, 12(Oct):2825–2830.

Sasa Petrovic, Miles Osborne, Richard McCreadie, Craig Macdonald, Iadh Ounis, and Luke Shrimpton. 2013. Can twitter replace newswire for breaking news? In *Seventh international AAAI conference on weblogs and social media*.

Takeshi Sakaki, Makoto Okazaki, and Yutaka Matsuo. 2010. Earthquake shakes twitter users: real-time event detection by social sensors. In *Proceedings of the 19th international conference on World wide web*, pages 851–860. ACM.

Jyoti Prakash Singh, Yogesh K Dwivedi, Nripendra P Rana, Abhinav Kumar, and Kawaljeet Kaur Kapoor. 2017. Event classification and location prediction from tweets during disasters. *Annals of Operations Research*, pages 1–21.

Abu Bakr Soliman, Kareem Eissa, and Samhaa R El-Beltagy. 2017. Aravec: A set of arabic word embedding models for use in arabic nlp. *Procedia Computer Science*, 117:256–265.

Zeynep Tufekci and Christopher Wilson. 2012. Social media and the decision to participate in political protest: Observations from tahrir square. *Journal of communication*, 62(2):363–379.